Microsoft .NET Glossary

This glossary will help explain the technologies, products and concepts associated with Microsoft® .NET.

Overview

.NET is Microsoft Corp.'s XML Web services platform, designed for building .NET experiences. It includes a programming model and development tools, enterprise servers, and a set of programmable XML Web services that will enable the next generation of applications. .NET consists of four primary components:

- .NET Framework and Visual Studio®.NET
- .NET Enterprise Server infrastructure
- Building-block services
- Software for smart devices

Definitions

ASP.NET is more than the next version of Active Server Pages. It is a compiled .NET Framework-based environment. Developers can author applications in any .NET Framework compatible language, including Visual Basic®, C# and JScript®. In addition, the entire .NET Framework platform is available to any ASP.NET application.

Building-block services are a user-centric set of XML Web services that move control of user data from applications to users. They enable personalized simplicity and consistency across applications, services and devices, while ensuring that the basis for all transactions is user consent. Building-block services include Microsoft Passport (for user identification) and services for message delivery, file storage, user-preference management, calendar management and other functions. Microsoft will offer a few building-block services in areas that are critical to the infrastructure of .NET, and a wide range of industry partners and developers will significantly expand the set of building-block services.

Common Language Infrastructure (CLI) is a vendor- and platform-neutral subset of the .NET Framework that enables companies to implement solutions anywhere in their enterprise. The CLI, developed by Microsoft, was submitted to the ECMA standards body for standardization in November 2000.

Common Language Runtime (CLR) is a high-performance execution engine. Code that targets the runtime and whose execution is managed by the runtime is referred to as “managed” code. Responsibility for tasks such as creating objects, making method calls and so on is delegated to the Common Language Runtime, which enables the runtime to provide additional services to the executing code. The Common Language Runtime actually has a role in both a component’s development time and execution time experiences. While the component is running, the runtime provides services such as memory management (including garbage collection), process management, thread management and security enforcement, as well as satisfying any dependencies that the component may have on other components.

Distributed computing is a programming model in which processing occurs in many different places (or nodes) around a network. Processing can occur wherever it makes the most sense, whether that is on a server, Web site, personal computer, handheld device or other smart device. In contrast to the two-node system — the client and the centralized server — prevalent today, Microsoft .NET uses distributed computing.

Extensible Markup Language (XML) is a universal method for representing structured information (spreadsheets, address books, purchase orders, technical information, etc.). It is especially well-suited to moving data in a distributed computing environment. With XML, developers can specify the structure of a document, for example, its title, its author or a list of related links. XML is the major building block for Web services, and its greatest strength is its ability to traverse any computing platform.

The **.NET Enterprise Server infrastructure** consists of the Microsoft Windows® operating system and the Microsoft .NET Enterprise Servers, a suite of infrastructure applications for building, deploying and operating applications and XML Web services. These servers, which offer deep support for XML, include the following:

- **Application Center 2000.** The deployment and management tool for high-availability Web applications that makes managing groups of servers as simple as managing one
- **BizTalk™ Server 2000.** The infrastructure to build and deploy integrated business processes and applications within the organization and with trading partners
- **Commerce Server 2000.** The application framework necessary to develop Web sites that optimize customer experience, encourage repeat business and forge tighter partner relationships

- **Exchange 2000 Server.** The e-mail, calendaring, messaging and collaboration solution for businesses of all sizes
- **Host Integration Server.** The follow-up release to Microsoft SNA Server that extends Microsoft Windows to other systems by providing application, data and network integration
- **Internet Security and Acceleration Server 2000.** The extensible, multilayer enterprise firewall and scalable high-performance Web cache
- **Microsoft Content Management Server.** The content management tool that enables fast deployment of scalable, dynamic Web sites and provides end users with a targeted, personalized experience tailored to an individual profile and browsing device
- **Microsoft Operations Manager.** The network systems management tool that provides event and performance management for the Windows 2000 Server family of operating systems and .NET Enterprise Server applications
- **Mobile Information 2001 Server.** Extends the reach of enterprise applications, enterprise data and intranet content into the realm of the mobile user
- **SharePoint™ Portal Server.** The flexible total portal solution that lets users easily find, share and publish information through document management and collaboration processes
- **SQL Server™ 2000.** The fully Web-enabled database and data-analysis package that provides core support for XML and the ability to query across the Internet and beyond the firewall
- **Systems Management Server.** The network systems management tool that provides a complete picture of a network's hardware and software resources

.NET experiences are dramatically more personal, integrated end-user computing interactions using connected XML Web services that are delivered through a new breed of smart devices. On a technical level, .NET experiences are a combination of XML Web services and, when appropriate, local application code.

The **.NET Framework** is the infrastructure for the overall .NET Platform. The common language runtime and class libraries (including Windows Forms, ADO.NET and ASP.NET) combine to provide services and solutions that can be easily integrated within and across a variety of systems, platforms and devices. The .NET Framework provides a fully managed, protected and feature-rich application execution environment; simplified development and deployment; and seamless integration with a wide variety of languages. A companion infrastructure, the .NET Compact Framework, is

a set of programming interfaces that enable developers to target mobile devices such as smart phones and personal digital assistants.

Peer-to-peer is a communication model in which computing devices — desktops, servers and other smart devices — link directly to each other. This can include smart-client-to-smart-client, smart-client-to-smart-server, and smart-server-to-smart-server connections. Microsoft .NET supports peer-to-peer as one of the manifestations of distributed computing. Peer-to-peer is the opposite of server-centric computing, in which all communication runs through a server. In the server-centric model, if an application wants to send data to another application, it would have to first contact a central server, which takes that data, reads the address of the recipient, and passes it along. .NET supports many forms of distributed computing, including peer-to-peer, in order to allow users to take advantage of the computing power that exists on their PCs and smart devices. This is often a more efficient application model and reduces bottlenecks associated with server-centric computing.

Simple Object Access Protocol (SOAP) is a lightweight and simple XML-based protocol that is designed to exchange structured and typed information on the Web. The purpose of SOAP is to enable rich and automated Web services based on a shared and open Web infrastructure. SOAP can be used in combination with a variety of existing Internet protocols and formats, including HTTP, SMTP and MIME, and can support a wide range of applications, from messaging systems to RPC.

Smart devices. .NET uses software for smart devices to enable PCs, laptops, workstations, smart phones, handheld computers, Tablet PCs, game consoles, Auto PCs and other smart devices to operate in the .NET universe. Microsoft is currently working on the following software products for smart devices: the Windows XP, Windows Millennium Edition (Windows Me),
Windows CE and Windows Embedded operating systems; the .NET Framework; and the .NET Compact Framework. A smart device has the following qualities:

- **Smart about the user.** It leverages a user’s .NET identity, profile and data to simplify the user experience and tailors notifications in response to the user’s presence or absence.
- **Smart about the network.** It is responsive to bandwidth constraints, provides support for both online and offline use of applications, and understands which services are available.
- **Smart about information.** It allows users to access, analyze and act on data anywhere, any time.
- **Smart about other devices.** It discovers and announces PCs, smart

devices, servers and the Internet, knows how to provide services to other devices, and is smart about accessing information from the PC.

- **Smart about software and services.** It presents applications and data optimally for form factor, enables input methods and connectivity appropriate for great end-user interaction, consumes Web services using XML, SOAP and UDDI, and is programmable and extensible by developers.

SOAP Discovery (DISCO) is a specification that provides a set of rules so that a human or development tool can automatically discover a service's WSDL description.

Unified classes. The .NET Framework provides a unified, object-oriented, hierarchical and extensible set of class libraries (i.e., APIs) so developers can use languages they are already familiar with.

Universal Description, Discovery and Integration (UDDI) is a directory, like the phone book, that provides a way to locate and register Web services on the Internet. It is a platform-independent, open framework for describing Web services, discovering businesses and integrating business Web services. It consists also of an operational registry, which is available today. The UDDI specification comprises three parts:

- White pages, which provide business contact information
- Yellow pages, which organize Web services into categories (e.g., credit card authorization services)
- Green pages, which provide detailed technical information about individual services

Visual Studio.NET provides a complete development environment for building applications and XML Web services on the Microsoft .NET platform. Using Visual Studio.NET, developers can create secure, scalable applications and XML Web services faster than ever before, using the language of their choice and leveraging existing systems and skills.

Web Service Description Language (WSDL) is an XML-based protocol that developers and development tools can use to represent the capabilities of a Web Service.

Windows XP is the new Windows desktop operating system for home and business computing. Windows XP delivers an intelligent, connected and dependable engine to power .NET experiences, based on a proven technical foundation.

XML Web services are applications made available over the Web via the Internet-standards XML, SOAP and UDDI. XML Web services are units of application logic that provide data and services to other applications. Applications access XML Web services via standard Web protocols and data formats such as HTTP, XML and SOAP, independent of how each XML Web service is implemented. XML Web services combine the best aspects of component-based development and the Web. They are a cornerstone of the Microsoft .NET programming model.

#########

Microsoft, Visual Studio, Visual Basic, JScript, Windows, BizTalk and SharePoint are either registered trademarks or trademarks of Microsoft Corp. in the United States and/or other countries.
The names of actual companies and products mentioned herein may be the trademarks of their respective owners.